

Excel を用いた区分的線形抵抗回路の全解探索に関する研究

Study on Algorithms for Finding All Solutions of Piecewise-Linear Resistive Circuits Using Excel

電気電子情報通信工学専攻 小山 大輝

Daiki KOYAMA

あらまし Microsoft Excel を用いた区分的線形抵抗回路のすべての解を求める簡単な方法を提案する。本手法は、区分的線形抵抗回路を記述する区分的線形方程式を混合整数計画問題に定式化し、圧倒的な普及度を誇る表計算ソフトである Microsoft Excel に搭載されている整数計画ソルバーを適用するものである。この方法は複雑なプログラミングを必要としないため実装が容易で、身近なソフトウェアである Excel を用いてすべての解（直流動作点、特性曲線）を求めることが可能となる。

1. まえがき

非線形回路のすべての解を求める効率的かつ実用的なアルゴリズムを確立することは、回路設計における重要な問題の一つである。この問題に対しては様々なアルゴリズムが提案されているが、その多くは複雑なプログラミングや専門的知識を必要とするものだった [1]–[4]。そのため、初心者や非専門家にとっては必ずしも実装容易性に優れた方法ではなかった。

ところで近年、数理計画の分野において整数計画法が急速に発展し、CPLEX [5] や SCIP [6] といった商用・非商用の整数計画ソルバーが開発されている。これらの整数計画ソルバーは 10 年前までは NP 困難という呪縛から「絶対に」解けないと考えられていた大規模な整数計画問題を実用的な計算時間で解けるようになり、現代社会に大きな影響を与えている。また、この 20 年間で整数計画ソルバーは平均で約 7 億倍高速になったと言われる。

本研究室では、このような整数計画法の飛躍的な発展に着目し、混合整数計画問題を CPLEX で 1 回解くだけで区分的線形抵抗回路のすべての解（直流動作点）を求める方法が提案された [7]。この方法は複雑なプログラミングを必要としないため実装が容易で、かつ非常に効率が良い。しかし、CPLEX は非常に高価な商用のソフトウェアである。CPLEX には無料のアカデミック版が存在するが、アカデミック環境にない研究者にとっては、これは大きな問題となる。また、非商用の整数計画ソルバー

の中で最も高速なソルバーとして SCIP が知られている。しかし SCIP は電気電子工学の分野ではあまり普及していないため、マニュアルが英語であることも含めて、扱いが容易ではない。

また最近、「ソルバーを使ったことが無いので英語のソルバーは不安である」、「小規模な問題しか解かないので、もっと手軽にコストをかけずに解析を行いたい」といった要望があがっている。初めて回路解析を行う人にとって、身近な日本語のソフトウェアで簡単に解析を行うことができれば、回路解析や非線形問題がより身近なものとなる。

本論文では、圧倒的な普及度を誇る表計算ソフトウェア Microsoft Excel に着目し、Excel に搭載されている整数計画ソルバーを用いた区分的線形抵抗回路の全解探索法を提案する。Excel は商用のソフトウェアであるが、Microsoft Office の中核をなすアプリケーションであるため、Windows パソコンのユーザーであれば簡単に入手可能で、利用しやすい。

本手法により、Excel の基本的な使い方に通じている人であれば、誰でも簡単に全解探索を行うことができる。

2. 整数計画法を用いた全解探索法

本章では、文献 [7] で提案された区分的線形抵抗回路のすべての解（直流動作点）を求める方法について説明する。 n 個の区分的線形抵抗を含む抵抗回路は、一般に次のような形の区分的線形方程式で記述することができる。

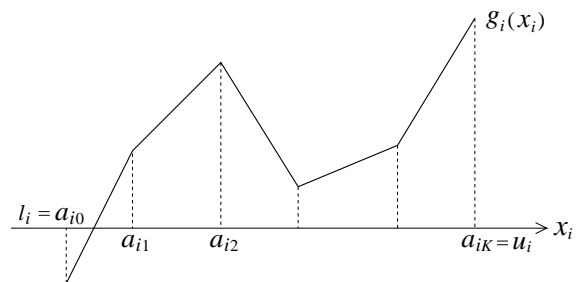


図 1 区分的線形関数

$$f(x) \triangleq Pg(x) + Qx - r = 0 \quad (1)$$

ただし, $x = (x_1, x_2, \dots, x_n)^T \in \mathbb{R}^n$ は区分的線形抵抗の枝電圧または枝電流を要素とする変数ベクトル, P, Q は回路の構造によって決まる $n \times n$ 定数行列, r は定数ベクトル, $g(x) = [g_1(x_1), g_2(x_2), \dots, g_n(x_n)]^T$ は図 1 に示すような区分的線形関数である. 式 (1) が線形方程式となるような領域を線形領域と呼ぶことにする.

文献 [7] では, 連続変数 λ_{ij} ($i = 1, 2, \dots, n; j = 0, 1, \dots, K$) と 0-1 変数 μ_{ij} ($i = 1, 2, \dots, n; j = 1, 2, \dots, K$) を導入することにより, 式 (1) を線形等式と線形不等式で表現できることが示されている. ここで, これらの線形等式・線形不等式の制約のもとで次のような混合整数計画問題を考える.

最大化: (任意の定数)

制約条件:

$$\begin{aligned} Py + Qx - r &= 0 \\ x_i &= \sum_{j=0}^K a_{ij} \lambda_{ij} \\ y_i &= \sum_{j=0}^K g_i(a_{ij}) \lambda_{ij} \\ \sum_{j=0}^K \lambda_{ij} &= 1 \\ \sum_{j=1}^K \mu_{ij} &= 1 \\ \lambda_{i0} &\leq \mu_{i1} \\ \lambda_{i1} &\leq \mu_{i1} + \mu_{i2} \\ &\vdots \\ \lambda_{iK-1} &\leq \mu_{iK-1} + \mu_{iK} \\ \lambda_{iK} &\leq \mu_{iK}, \quad i = 1, 2, \dots, n \end{aligned} \quad (2)$$

式 (2) の制約条件と式 (1) が等価であることは容易に確認できる. すなわち, 式 (2) を満たすすべての解を求めることにより, 式 (1) のすべての解を得ることができる. また, 制約条件を満たす 0-1 変数 μ_{ij} の組合せと線形領域の間には 1 対 1 対応が存在する.

文献 [7] では, 式 (2) を解くための整数計画ソルバーとして, 現時点で最も高速な商用ソフトウェアの一つである CPLEX [5] を用いている. CPLEX を用いることの大きな利点は, CPLEX には解プールという機能があり, この機能を利用することによってより簡単な全解探索が可能になることである. 解プールとは, 混合整数計画問題の制約条件を満たす解を求め, 保存する機能である. すなわ

ち, 解プールの機能を用いることで, 混合整数計画問題 (2) を 1 回解くだけですべての解を求めることができる.

3. 提案手法

CPLEX を使用する際の欠点は, 非アカデミックユーザーには高価なことである. 本研究では, Excel を用いて混合整数計画問題 (2) を解くことを考える. しかし, Excel に搭載されているソルバーには解プール機能が存在しないため, 一度に解を一つしか求められない. そこで, 解プール機能を用いずすべての解を求めるための工夫を考える.

Excel ソルバーで式 (2) を解いた結果, 解 α が得られたものとする. また α が存在する線形領域を R とする. 式 (2) では, 各 $i = 1, 2, \dots, n$ に対して一つの μ_{ij} だけが 1 となり, 他はすべて 0 になる. すなわち

$$\mu_{ij} = \begin{cases} 1 & (j = k_i) \\ 0 & (j \neq k_i) \end{cases}, \quad i = 1, 2, \dots, n \quad (3)$$

となる. ただし k_i は各 i に対して μ_{ik_i} の値が 1 となる添え字を表す.

一つの線形領域に対する μ_{ij} の組合せは一意的に定まり, 他の線形領域では同じ組合せは現れない. したがって, もし解が得られた線形領域 R において式 (3) が成立したとすると, 他の線形領域では μ_{ik_i} ($i = 1, 2, \dots, n$) の値がすべて 1 になることはなく, 少なくとも一つは 0 となる. ここで, 次のような制約式を考える.

$$\sum_{i=1}^n \mu_{ik_i} \leq n - 1 \quad (4)$$

明らかに R では $\sum_{i=1}^n \mu_{ik_i} = n$ となり, 他の線形領域では $\sum_{i=1}^n \mu_{ik_i} \leq n - 1$ となるので, 式 (4) は「既に解が得られた線形領域のみを解析対象から外す制約条件」として利用することができる. すなわち, 解を一つ求める度に制約条件に式 (4) を追加しながら, 解が得られなくなるまで式 (2) を解くことにより, 式 (1) のすべての解を確実に求めることができる.

ここで, 具体例として図 2 を用いて本手法を説明する. ただし図 2 において $n = 2, K = 3$ であり, $\alpha^1, \alpha^2, \alpha^3$ は解である.

式 (2) を解くと, 解 α^1 が得られたとする. この時, 解 α^1 が得られた線形領域では 0-1 変数 μ_{ij} は次のように

なる。

$$\begin{aligned} \mu_{11} = 0, \quad \mu_{12} = 0, \quad \mu_{13} = 1 \\ \mu_{21} = 0, \quad \mu_{22} = 1, \quad \mu_{23} = 0. \end{aligned}$$

解 α^1 を含む線形領域では $\mu_{13} + \mu_{22} = 2$ となるので、式 (4) より線形不等式

$$\mu_{13} + \mu_{22} \leq 1, \quad (5)$$

を式 (2) の制約条件に追加することで、解 α^1 を含む線形領域を解析対象から取り除くことができる。そのため、次に式 (5) を追加した式 (2) を解くことで、2つ目の解 α^2 を得ることができる。解 α^2 が得られた線形領域では 0-1 変数 μ_{ij} は次のようになる。

$$\begin{aligned} \mu_{11} = 0, \quad \mu_{12} = 0, \quad \mu_{13} = 1 \\ \mu_{21} = 0, \quad \mu_{22} = 0, \quad \mu_{23} = 1. \end{aligned}$$

先程と同様に解 α^2 を含む線形領域では $\mu_{13} + \mu_{23} = 2$ となるので、式 (4) より線形不等式

$$\mu_{13} + \mu_{23} \leq 1, \quad (6)$$

を式 (2) の制約条件に追加することで、解 α^2 を含む線形領域が解析対象から取り除かれる。そのため、次に式 (5)、式 (6) を追加した式 (2) を解くことで、3つ目の解 α^3 を得ることができる。そして、式 (5)、式 (6) に加えて解 α^3 を解析対象から取り除くための線形不等式

$$\mu_{12} + \mu_{21} \leq 1, \quad (7)$$

を追加した式 (2) を解くと、実行可能領域が存在しないという情報を得ることができるため、解析が完了となる。

4. すべての特性曲線を求める方法

本章では、提案手法を拡張し、すべての特性曲線を求める方法を提案する。ここでは、図3に示すような n 個の区

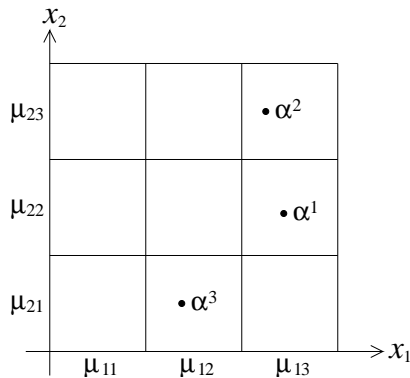


図2 提案手法の具体例

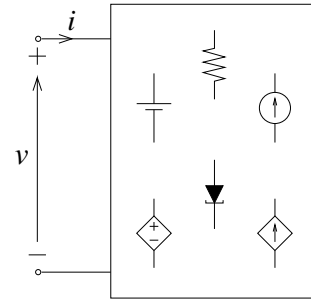


図3 1ポート区分的線形抵抗回路

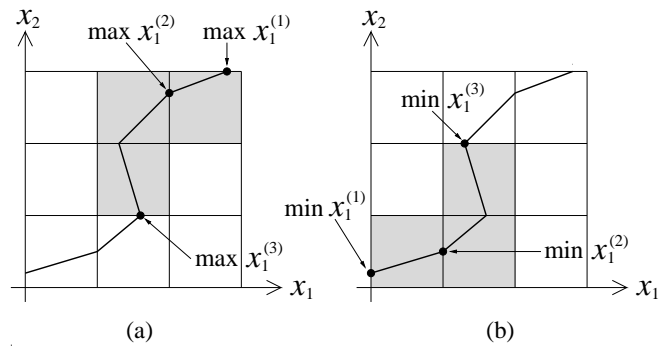


図4 最大化・最小化によって得られる折れ点の順序 (x_1 の上付き文字は順序を表す)

分的線形抵抗を含む1ポート回路の全ての特性曲線（駆動点特性曲線あるいは伝達特性曲線）を求める問題を考える。ポートの枝電圧を v 、枝電流を i とすると、図3の回路は一般に $n+2$ 個の変数と $n+1$ 個の区分的線形方程式によって $P[g_1(x_1), \dots, g_n(x_n), i]^T + Q(x_1, \dots, x_n, v)^T - r = 0$ と記述することができる。ただし $(x_1, x_2, \dots, x_n, v, i)^T \in \mathbb{R}^{n+2}$ は変数ベクトル、 P, Q は $(n+1) \times (n+1)$ 定数行列、 $r = (r_1, r_2, \dots, r_{n+1})^T \in \mathbb{R}^{n+1}$ は定数ベクトル、 $g_i(x_i)$ ($i = 1, 2, \dots, n$) は区分的線形抵抗の特性を表す区分的線形関数である。

ここで、すべての特性曲線を求めるため、式 (2) の目的関数を（任意の定数）から任意の変数 x_i (v や i など) とし、これを最大化・最小化する混合整数計画問題を考える。そして、最大化・最小化問題を提案手法を用いて解く。最大化問題を解くことにより、図4(a)に示す順序で区分的線形特性曲線の折れ点を得られる。そして最小化問題を解くことにより、図4(b)に示す順序で折れ点を得られる。すなわち、特性曲線が通過する各線形領域に対して、その領域における特性曲線の両端点（その線形領域を表す 0-1 変数 μ_{ij} とともに）得られる。これらの端点を結ぶことにより、すべての特性曲線を求めることができる。

表 1 例 2 の解

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
1	0.2169	0.2794	0.3419	0.4044	0.4669	0.5294	0.5919	2.0037	0.7169	0.7794
2	0.2191	0.2816	0.3441	0.4066	0.4691	0.5316	0.5941	1.9471	0.7191	0.7816
3	0.1735	0.2360	0.2985	0.3610	0.4235	0.4860	0.5485	0.6110	1.8116	2.0732
4	0.1654	0.2279	0.2904	0.3529	0.4154	0.4779	0.5404	0.6029	2.0131	2.0663
5	0.2172	0.2797	0.3422	0.4047	0.4672	0.5297	0.5922	0.6547	2.0572	0.7797
6	0.2064	0.2689	0.3314	0.3939	0.4564	0.5189	0.5814	0.6438	2.0480	1.0490
7	0.2176	0.2800	0.3426	0.4051	0.4676	0.5301	0.5926	0.6551	0.7176	2.1107

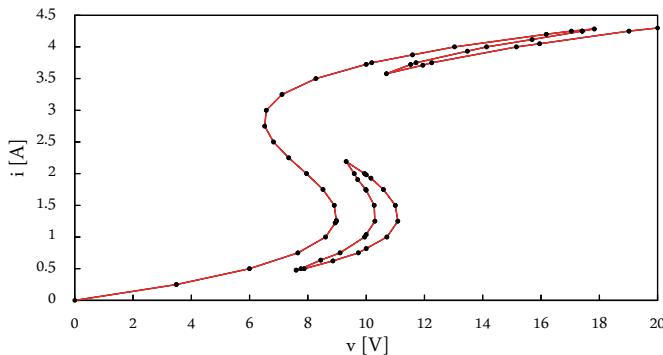


図 5 特性曲線 (例 3)

5. 数値例

本章ではいくつかの数値例を示す。なお使用計算機は HP Pavilion Slimline 400-420jp (CPU: Intel Core i7-4790 3.6GHz) で、Microsoft Excel 2016 を使用した。

例 1: 文献 [2] の図 3 に示されている 2 つのトンネルダイオードを含む回路に対し、区分的線形関数の線分数 $K = 10$ として本手法を適用した結果、9 個の解を得られた。計算時間は 0.7 秒であった。

例 2: 文献 [3] の図 9 に示されている 10 個のトンネルダイオードを含む回路に対し、 $K = 3$ として本手法を適用した結果、表 1 に示す 7 個の解を得られた。計算時間は 4 秒であった。

例 3: 文献 [8] の図 1 に示されている 1 ポート区分的線形抵抗回路に対し、 $K = 20$ として本手法を適用した結果、図 5 に示す駆動点特性曲線を得られた。特性曲線を構成する線分数は 57 であった。

参考文献

- [1] L.O. Chua and R.L.P. Ying, "Finding all solutions of piecewise-linear circuits," *Int. J. Circuit Theory Appl.*, vol.10 no.3, pp.201–229, July 1982.
- [2] K. Yamamura and M. Ochiai, "An efficient algorithm for finding all solutions of piecewise-linear resistive circuits," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol.39, no.3,

pp.213–221, March 1992.

- [3] K. Yamamura and T. Ohshima, "Finding all solutions of piecewise-linear resistive circuits using linear programming," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol.45, pp.434–445, April 1998.
- [4] K. Yamamura, K. Suda, and N. Tamura, "LP narrowing: A new strategy for finding all solutions of nonlinear equations," *Applied Mathematics and Computation*, vol.215, no.1, pp.405–413, Sept. 2009.
- [5] IBM, IBM ILOG CPLEX Optimization Studio, CPLEX User's Manual, Version 12, Release 6, http://www-01.ibm.com/support/knowledgecenter/SSSA5P_12.6.2/ilog.odms.studio.help/pdf/usrcplex.pdf
- [6] SCIP (Solving Constraint Integer Programs), <http://scip.zib.de/>
- [7] K. Yamamura and N. Tamura, "Finding all solutions of separable systems of piecewise-linear equations using integer programming," *J. Computational and Applied Mathematics*, vol.236, no.11, pp.2844–2852, May 2012.
- [8] A. Ushida and L.O. Chua, "Tracing solution curves of nonlinear equations with sharp turning points," *Int. J. Circuit Theory Appl.*, vol.12, no.1, pp.1–21, Jan. 1984.

研究業績

国際会議

- [1] S. Ishiguro, D. Koyama, and K. Yamamura, "Statistical tolerance analysis of nonlinear circuits using integer programming and set-valued functions with probability distribution," *Proc. 2014 IEEE Workshop on Nonlinear Circuit Networks*, pp.14–17, Tokushima, Japan, Dec. 2014.
- [2] D. Koyama and K. Yamamura, "Finding all solutions of piecewise-linear resistive circuits using Excel," *Proc. 2015 IEEE Workshop on Nonlinear Circuit Networks*, pp.38–41, Tokushima, Japan, Dec. 2015.
- [3] K. Yamamura and D. Koyama, "Finding all solutions of piecewise-linear resistive circuits using Excel," *Proc. IEEE Asia Pacific Conference on Circuits and Systems*, pp.228–231, Jeju, Korea, Oct. 2016.
- [4] K. Yamamura, D. Koyama, and S. Sato, "Finding all solution sets of piecewise-linear interval equations using integer programming," *Proc. 2016 IEEE Workshop on Nonlinear Circuit Networks*, pp.28–31, Tokushima, Japan, Dec. 2016.

国内会議

- [1] 小山大輝, 石黒俊, 山村清隆, "Excel を用いた区分的線形回路の全解探索," 2015 年電子情報通信学会ソサイエティ大会講演論文集, A-2-20, Sept. 2015.